

VYSOKÁ ŠKOLA BÁŇSKÁ – TECHNICKÁ UNIVERZITA OSTRAVA
EKONOMICKÁ

KATEDRA APLIKOVANÉ INFORMATIKY

Tvorba webové aplikace „Hudební e-shop“

Implementation of Web Application „Online Music Store“

Student: Aleksandar Nikolaev Kolev

Vedoucí bakalářské práce: Ing. Vítězslav Novák, Ph.D.

Ostrava 2019

Zadání bakalářské práce

Student: **Aleksandar Nikolaev Kolev**
Studijní program: **B6209 Systémové inženýrství a informatika**
Studijní obor: **6209R017 Informatika v ekonomice**
Téma: **Tvorba webové aplikace „Hudební e-shop“**
Implementation of Web Application „Online Music Store“
Jazyk vypracování: **čeština**

Zásady pro vypracování:

1. Úvod
2. Teoretické nástroje pro tvorbu webové aplikace
3. Analýza požadavků a dosaženého stavu
4. Návrh a implementace webové aplikace
5. Závěr

Seznam použité literatury

Seznam zkratk

Prohlášení o využití výsledků bakalářské práce

Seznam příloh

Přílohy

Seznam doporučené odborné literatury:

PEACOCK, Michael. *Programujeme vlastní e-shop v PHP 5*. Brno: Computer Press, 2011.

ISBN 978-80-251-3181-7.

SKLAR, David. *PHP 7: praktický průvodce nejrozšířenějším skriptovacím jazykem pro web*. Brno: Zoner Press, 2018. ISBN 978-80-7413-363-3.

WELLING, Luke and Laura THOMSON. *PHP and MySQL web development*. Fifth edition. Hoboken, NJ: Addison-Wesley, 2017. ISBN: 9780321833891.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Vítězslav Novák, Ph.D.**

Datum zadání: **23.11.2018**

Datum odevzdání: **10.05.2019**



Ing. Petr Rozehnal, Ph.D.
vedoucí katedry



prof. Dr. Ing. Zdeněk Zmeškal
děkan fakulty

Prohlašuji, že jsem celou práci, včetně všech příloh, vypracoval samostatně.

V Ostravě dne 20. 6. 2019



.....
Aleksandar Nikolaev Kolev

Zde bych chtěl poděkovat panu Ing. Vítězslavu Novákovi, Ph.D. za odborné vedení mé bakalářské práce, jeho cenné rady, připomínky a čas, který mi věnoval.

Obsah

1 Úvod.....	5
2. Teoretické nástroje pro tvorbu webové aplikace	6
2.1 Historie Internetu a jazyku HTML	6
2.2 HTML5	6
2.3 CSS.....	8
2.3.1 Stavba pravidla stylu.....	9
2.3.2 Formátování textu.....	9
2.3.3 Barvy.....	10
2.4 PHP.....	11
2.5 MySQL a phpMyAdmin	12
2.5.1 Relace v databázích	12
2.5.2 Příkazy jazyka SQL	12
2.5.3. Klauzule jazyka SQL	13
2.5.4 phpMyAdmin	15
2.6 JavaScript.....	16
2.7 Framework a Bootstrap.....	16
2.8. Elektronické obchodování	18
2.8.1 Přehled elektronického obchodování.....	18
3. Analýza požadavků a dosaženého stavu.....	20
3.1 E-shop „Bontonland“	20
3.2 E-shop „Music Records“	21
3.3 E-shop „EMP“.....	22
3.4. Požadavky	23
4. Návrh a implementace webové aplikace	24
4.1. Databáze.....	24
4.1.1. Relační model.....	24
4.1.2. Tabulky a vztahy	24

4.2. Komunikace mezi databází a webovou stránkou	27
4.3 Webové stránky.....	28
4.3.1 Responzivní design stránky.....	28
4.3.2 Header a Footer	28
4.3.3 Přehled novinek na hlavní stránce	28
4.3.4. Prohlížení zboží	29
4.3.5. Rychlé nahlížení do popisu zboží	31
4.3.6. Vyhledání zboží.....	31
4.3.7. Přihlášení a registrace	31
4.3.8. Zobrazení podrobnosti o zboží	33
4.3.9. Přidání zboží do košíku.....	34
4.3.10. Hodnocení zboží	34
4 3.11. Košík uživatele	35
4 3.12. Objednávka a e-mail	35
4.4 Administrační rozhraní	36
5. Závěr.....	40
Seznam použité literatury	41
Seznam zkratk	42
Prohlášení o využití výsledků bakalářské práce	3

1 Úvod

V dnešní moderní době, které dominují zejména technologie a Internet máme možnost z pohodlí domova realizovat více činnosti než kdykoliv dříve. Jednou z nich je nakupování prostřednictvím e-shopů neboli internetových obchodů. Tento způsob nákupu je oblíbený jak ze strany zákazníka (především kvůli rychlosti), tak i ze strany podniku. Založením vlastního e-shopu můžeme například snížit náklady na provoz tím, že zrušíme kamennou prodejnu a snížíme četnost personálu.

Realizovat obchod bez přímého kontaktu prodávajícího a klienta je věc, která by byla před 50 lety zcela nepředstavitelná. Jsme však svědky případů, kde se podniky bez kamenné prodejny (jako například americký Amazon) stávají dominantou na trhu. Na tomto příkladu je důležité si všimnout také různorodost zboží, které obchod nabízí. Je to jedna z obrovských výhod e-shopu, jelikož problémy jako velikost kamenné prodejny a specializace personálu jsou mnohem nákladnější na překonání, než rozšíření sortimentu na internetovém obchodu.

V druhé kapitole jsou popsány a vysvětleny teoretické nástroje, které byly v rámci tvorby aplikace použity. V rámci další kapitoly je zanalyzována konkurence a jsou stanovené požadavky na konečný výsledek práce. Ve čtvrté kapitole jsou podrobně vysvětleny funkčnost aplikace a způsoby, použité pro její realizaci.

Cílem mé bakalářské práce je vytvořit hudební e-shop pro prodej diskografií interpretů a kapel různých žánrů. Aplikace je určena mému budoucímu podniku, který plánuju po dokončení studia zahájit. Budou stanoveny realistické požadavky tak, aby se konečný výsledek přiblížil skutečným e-shopům v této odvětví a mohl být využitelný v praxi. Aplikace se bude skládat z databáze a webovou aplikaci. Administrátor bude mít k dispozici rozhraní pro správu databáze. Uživatel bude mít možnost si zboží prohlížet, a filtrovat jej různých kritérii. Přičemž mu bude umožněno si zvolené zboží objednat a poté jej také ohodnotit.

2. Teoretické nástroje pro tvorbu webové aplikace

V této kapitole se budeme věnovat jazykům a technologiím, které budou použity při návrhu a implementaci tohoto e-shopu.

2.1 Historie Internetu a jazyku HTML

První myšlenky na vytvoření počítačové sítě se objevily na začátku šedesátých let v USA. Šlo o nápad, jak propojit vojenské, vládní a vědecko-výzkumné počítače jako rovnocenné uzly, které by mohly vysílat a přijímat zprávy. Později v roce 1969 vznikla síť ARPANET, která propojovala 4 univerzity - University of California Los Angeles, Stanford Central Research Institute, University of California Santa Barbara a University of Utah. Postupně se na ní připojovalo stále více sítí, díky čemuž se začala vést diskuze o novém názvu v této době již globální sítě – tehdy vznikl pojem Internet. V roce 1990 byla síť ARPANET zrušena, přičemž v té době bylo v síti propojeno více než 100 000 počítačů. O rok později byla v Evropské organizaci pro jaderný výzkum CERN nasazena internetová služba WWW neboli World Wide Web, což je informační prostor, tvůrcem kterého je britský Tim Berners-Lee. V dnešní době působí jako profesor na Oxfordské univerzitě a jako ředitel konsorcia W3C, které dohlíží na pokračující vývoj webu.

Jazyk HTML (Hyper Text Markup Language) je značkovací jazyk, který se používá pro tvorbu webových stránek. Vznikl v roce 1990 a jeho autoři jsou Tim Berners-Lee a Robert Cailliau. HTML kód je kombinací prostého textu a značek. Smysl těchto značek (neboli tagů jak si jím také říká) je v oddělování a vyznačování částí textu.

2.2 HTML5

Nejaktuálnější verze jazyka HTML je verze 5.2 (14. prosinec 2017). Oproti předchozí verzi 4 byly například přidány nové tagy jako canvas, video a audio. Dále došlo k zjednodušení syntaxe, což se projevilo například u elementu DOCTYPE (definován na začátku každého HTML dokumentu), jehož deklarace je v nové verzi kratší a srozumitelnější: `<!DOCTYPE html>`. V této práci bude použita již výše zmíněná verze jako základ pro založení webu a definování jeho elementů jako např. formuláře, tabulky, atd.

Samotná syntaxe jazyka je velmi intuitivní a lehce zapamatovatelná, proto se HTML5 těší oblibě mezi začátečníky i pokročilými vývojáři. V současné době ho používá víc než 70% všech webů na světě. Na následujícím obrázku si ukážeme jednoduchý příklad HTML5 kódu.

```
<!DOCTYPE html>

<html>
  <head>
    <title>Simple Web Page</title>
    <meta charset="UTF-8">
  </head>

  <body>
    <header>
      <nav id="menu">
        <a href="#">Option One</a>
        <a href="#">Option Two</a>
        <a href="#">Option Three</a>
      </nav>
    </header>

    <div id="main">
      <h1>Heading</h1>
      <article></article>
      :
    </div>

  </body>
</html>
```

Obrázek 2.1. Ukázka HTML5 kódu (zdroj: Vlastní)

Jazyk HTML5 je podporován částečně či skoro úplně všemi webovými prohlížeči dostupné pro desktopové počítače a mobilní zařízení. Na webu html5test.com lze otestovat podporu právě používaného prohlížeče a navíc také porovnat výkon prohlížečů mezi sebou dokonce i s výběrem verze. Z maximálních 555 bodů ohodnocení podpory se aktuální verze nejpoužívanějších počítačových browserů umístili takto:

- Microsoft Edge, verze 18 (nástupce starého známého Exploreru) – 496 bodů
- Google Chrome, verze 68 – 528 bodů
- Mozilla Firefox, verze 60 – 497 bodů
- Opera, verze 45 – 518 bodů
- Safari, verze 11.2 – 477 bodů

2.3 CSS

Cascading Style Sheets neboli CSS je jazyk, který určuje způsob zobrazení elementů z jazyka HTML (písmo, barvy, rozmístění prvků, atd.). Hlavním smyslem kaskádových stylů je oddělení vzhledu dokumentu od jeho struktury a obsahu. Použitím tzv. selektorů a bloků deklarací nastavujeme vlastnosti elementů, deklarovaných v rámci HTML kódu. Autorem prvotního návrhu byl Håkon Wium Lie, který jejich koncept publikoval v roce 1994 v době, kdy pracoval pro organizaci CERN.

První verze jazyka CSS se objevily až několik let po vzniku jazyka HTML, přičemž oficiálně se tento jazyk prosadil v roce 1996. Vztah mezi jazykem CSS3 a jeho staršími verzemi je analogický ke vztahu jazyka HTML5 s jeho předchůdci – jazyk CSS3 je přirozeným rozšířením svých starších verzí. Jazyk CSS3 je mnohem mocnější než jeho předchůdci. Zavádí totiž několik vizuálních efektů, zaoblené rohy a přechody. (Castro, 2012)

Zatímco pomocí jazyka HTML definujeme význam obsahu a udělujeme svým webovým stránkám základní strukturu, s jazykem CSS (Cascading Style Sheets) specifikujeme vzhled. Šablona stylů je obyčejný textový soubor, jenž obsahuje jedno nebo více pravidel, která prostřednictvím svých vlastností a hodnot určují, jak by se měly určité elementy zobrazovat. Jazyk CSS nabízí vlastnosti pro formátování textu; například velikost a barvu písma, dále pak vlastnosti pro definici rozvržení, což jsou různé vlastnosti pro umísťování elementů, a také vlastnosti pro řízení tisku. Jazyk CSS poskytuje rovněž několik dynamických vlastností, které umožňují elementům objevovat se a skrývat. Tyto vlastnosti jsou užitečné pro tvorbu rozvíracích seznamů a jiných interaktivních komponent. (Castro, 2012)

```
body{ background-color: black; }

h1{ text-align: center;
    font-size: 120%;
    font-family: Verdana;
}

a{
    color: white;
    padding: 10px 25px;
    text-decoration: none;
    display: inline-block;
}
```

Obrázek 2.2. Ukázka CSS3 kódu (zdroj: Vlastní)

2.3.1 Stavba pravidla stylu

Každé pravidlo stylu v šabloně stylů se skládá ze dvou hlavních částí:

- selektoru,
- deklaračního bloku, jenž se skládá z jedné nebo více dvojic vlastnost-hodnota, přičemž každá tato dvojice tvoří tzv. deklaraci, jež specifikuje, co by se s vybranými elementy mělo dít.

Pravidlo stylu sestavíme následovně:

1. Napíšeme selektor, což je výraz, jenž identifikuje element/y, které chceme stylvat.
2. Zahájíme deklarační blok zápisem znaku { (levou složenou závorkou).
3. Vložíme deklaraci vlastnost: hodnota, přičemž vlastnost představuje název vlastnosti jazyka CSS, která popisuje určitý aspekt vzhledu a hodnota je jedna z možných hodnot pro danou vlastnost.
4. Opakujeme krok 3, kolikrát to bude nutné. Obvykle zapisujeme každou dvojici vlastnost: hodnota (deklaraci) na samostatný řádek
5. Napíšeme znak }, čímž ukončíme jak deklarační blok, tak celé pravidlo stylu. (Castro, 2012)

Selektor rozhoduje o tom, na jaké elementy se dané pravidlo stylů aplikuje. Když kupříkladu budeme chtít přidělit všem elementům p písmo Times o velikosti 12 pixelů, musíme vytvořit selektor, který vybere jen elementy p a ostatní elementy nechá na pokoji. Kdybychom potřebovali ovlivnit vzhled prvního odstavce v každé sekci, museli bychom sestavit o něco komplikovanější selektor, jenž by identifikoval pouze ty elementy p, které jsou prvními potomky jednotlivých sekcí. (Castro, 2012)

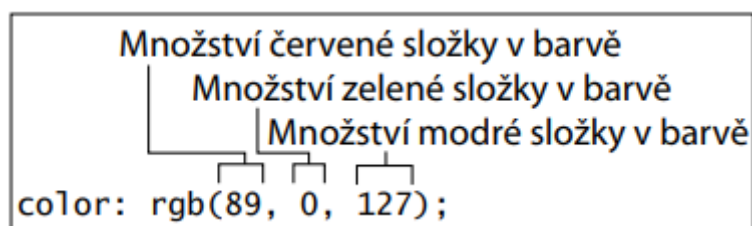
2.3.2 Formátování textu

S pomocí jazyka CSS můžeme měnit písmo, velikost, tučnost, naklonění, výšku řádku, barvu popředí a pozadí, velikost mezer a zarovnání textu. Můžeme se rozhodovat, jestli text podtrhneme, přeškrtneme, převedeme jeho písmena na velká, malá nebo na kapitálky. Rovněž můžeme tyto změny aplikovat na celý dokument, nebo dokonce celé webové stránky, a to jen s několika řádky kódu. (Castro, 2012)

Barvy je možné definovat u vlastností jazyka CSS několika způsoby. Zaprvé – hodnotou může být předdefinované jméno barvy. Jazyk CSS3 definuje seznam 16 základních jmen barev (viz obrázek 7.10), k nimž přidává dalších 131 jmen, aby vyhověl 147 jménům barev jazyka SVG 1.0. Kompletní seznam je k dispozici na internetové adrese <http://www.w3.org/TR/css3-color/#svg-color>. (Castro, 2012)

Formát RGB

Vlastní barvu můžeme sestavit tak, že definujeme množství červené, zelené a modré složky (odtud pochází název RGB – Red Green Blue, což jsou v češtině barvy červená, zelená a modrá). Každé této barevné složce můžeme přiřadit číslo od 0 do 255, procento nebo šestnáctkovou reprezentaci čísla. Kdybychom kupříkladu chtěli vytvořit tmavě purpurovou barvu, použili bychom číslo 89 pro červenou složku, nulu pro zelenou složku a 127 pro modrou složku. Takovou barvu bychom mohli zapsat pomocí kódu `rgb(89, 0, 127)`. (Castro, 2012)



10

2.4 PHP

Historie PHP začala vcelku nenápadně a jako u většiny velkých věcí zde svůj díl hrálo i to, že se objevilo ve vhodném místě a v pravý čas. Někdy kolem roku 1995 vydal Rasmus Lerdorf sadu skriptů v jazyce Perl pod názvem PHP/FI. Tato sada skriptů se stala velmi používanou, a tak roce 1997 vzniklo PHP/FI verze 2, které již bylo psáno v jazyce C. V roce 1997 se Andi Gutmans a Zeev Suraski pokoušeli použít PHP/FI pro vývoj komerčních aplikací a shledali jej jako poddimenzované pro tyto účely. Rozhodli se jej proto kompletně přepsat a výsledek označili jako PHP verze 3. V zimě 1998 se začalo pracovat na přepsání jádra PHP, aby se zvýšil výkon pro složité aplikace. Vzniklo tak nové jádro nazvané Zend podle počátečních písmen svých tvůrců. V roce 2000 se pak objevilo PHP verze 4 s tímto novým jádrem. Po vylepšení jádra Zend a doplnění dalších možností se v roce 2004 objevilo PHP verze 5. Hlavní revoluci verze 5 je podstatné vylepšení možností objektově orientovaného programování. (Ponkrác, 2007)

PHP je velmi populárním jazykem, a protože sám o sobě není frameworkem, můžeme s jeho pomocí snadno vytvořit náš vlastní framework, tak jak potřebujeme. Volba programovacího jazyka je zpravidla otázkou osobních preferencí. Většina moderních webových hostingů podporuje PHP a MySQL, a přestože i další jazyky, jako je např. Ruby on Rails získávají na popularitě, jejich podpora ze strany hostingů není ani zdaleka tak běžná. (Peacock, 2011)

Na dalším obrázku je kus PHP kódu, kde je definovaná proměnná, je jí přiřazena hodnota a potom zrušená.

```
$str_isset = "The isset example";
$bol_isset = isset($str_isset);

If ($bol_isset){
    echo "The variable is set </BR>";
}
else {
    echo "The variable is not set </BR>";
}
//Using unset function
unset ($str_isset);
$bol_isset = isset($str_isset);
If ($bol_isset){
    echo "The variable is set";
}
else {
    echo "The variable is not set";
}
```

Obrázek 2.4. Ukázka PHP kódu (zdroj: Vlastní)

2.5 MySQL a phpMyAdmin

“Structured query language” neboli SQL je strukturovaný dotazovací jazyk, který umožňuje práci s daty v relačních databázích. Obsahuje příkazy, pomocí kterých můžeme data definovat, zobrazovat, manipulovat s nimi, provádět transakce a navíc také řídit přístupová práva uživatelů. MySQL je systém řízení báze dat, ve kterém komunikace probíhá právě pomocí jazyka SQL. Samotný jazyk MySQL se velmi podobá jazyku SQL s malými odchylkami v syntaxi. Jeho tvůrcem je švédská firma MySQL AB, ale v současné době je jeho vlastníkem společnost Oracle Corporation.

2.5.1 Relace v databázích

Abychom mohli mezi databázovými tabulkami vytvářet logické vazby, tak je třeba mezi nimi vytvořit vztahy (relace). Druhy relace dělíme takto:

- Relace 1:1 – nejjednodušší vazba (jednomu záznamu v tabulce odpovídá právě jeden záznam v druhé tabulce), typickým příkladem v našem případě může být zákazník a košík
- Relace 1:N – nejpoužívanější vztah, kde jeden záznam v první tabulce odpovídá několika záznamům v druhé tabulce, například osoba a objednávka. Jedna osoba může mít více objednávek
- Relace M:N – složitější vztah, kde více záznamů v první tabulce odpovídá více záznamům v druhé tabulce. Je řešen zavedením další tabulky, která bude mít s původními tabulkami relaci 1:N

V dalších podkapitolách se blíže seznámíme s jazykem SQL.

2.5.2 Příkazy jazyka SQL

Jazyk DDL (Data Definition Language)

Určen pro tvorbu objektů jako např. tabulky nebo sestavy. Umožňuje rovněž měnit jejich strukturu či odstraňovat je. Jeho příkazy jsou:

- CREATE – pro definice databází, tabulek, pohledy apod.
- ALTER - pro změnu struktury objektů (např. definice nového sloupce, v rámci již existující tabulky)
- DROP – pro mazání objektů, dá se použít i v kombinaci s příkazem ALTER, např. v situaci kdy chceme odstranit sloupec z nějaké tabulky

Jazyk DML (Data Manipulation Language)

Příkazy tohoto jazyka umí manipulovat s daty. Tím je myšleno vkládání či výběr dat, ale zároveň aktualizace a mazání záznamů. Tento jazyk obsahuje čtyři hlavní příkazy:

- SELECT – pro výběr dat
- INSERT – pro vkládání dat do tabulky
- UPDATE – pro editace záznamů
- DELETE – mazání záznamů

2.5.3. Klauzule jazyka SQL

Při psaní SQL dotazů se používají tzv. klauzule, které umí položky seřadit, seskupovat a filtrovat. Každý SQL dotaz začíná příkazem SELECT. Po něm obvykle následují názvy sloupců, ale můžeme rovněž používat agregační funkce:

- SUM – určená pro součet hodnot ve sloupci
- MIN – zjistí minimální hodnotu ve sloupci
- MAX – zjistí maximální hodnotu ve sloupci
- COUNT – spočítá počet záznamů ve sloupci
- AVG – spočítá aritmetický průměr hodnot ve sloupci

Klauzule FROM

Pomocí klauzule FROM specifikujeme místo, kde se požadované informace nacházejí. Informace mohou být umístěny v jedné tabulce, ale mohou být roztroušeny ve více tabulkách, sestavách apod. (Lacko, 2005)

Klauzule WHERE

Klauzule WHERE určuje podmínky, jež přesně specifikují podmnožinu záznamů, které potřebujeme vybrat. Podmínkou můžeme stanovit zásady výběru dat z více databázových tabulek. V klauzuli WHERE můžeme použít například operátory pro porovnávání číselných hodnot:

- Rovnost =
- Nerovnost <>

- Větší >
- Menší <
- Větší nebo rovný >=
- Menší nebo rovný <=
- Patří do intervalu – BETWEEN
- Nepatří do intervalu – NOT BETWEEN

U znakových datových typů zpravidla pracujeme s porovnávacími operátory poněkud odlišným způsobem, přičemž používáme rovněž operátor LIKE. Tento operátor používá různé zástupné znaky (například znak procenta nahrazuje libovolný text). (Lacko, 2005)

Pokud chceme zadat několik podmínek WHERE pro ten samý atribut, tak se používá příkaz OR.

Další operátor, který se dá použít v klauzuli WHERE je IN (případně NOT IN). Pomocí něho můžeme v podmínce specifikovat více hodnot zároveň. Je taky i alternativní varianta pro příkaz OR.

Pokud chceme získat položky s prázdnou hodnotou, používáme operátor IS NULL a naopak pro položky, které nejsou prázdné IS NOT NULL.

Klauzule GROUP BY

Klauzule „GROUP BY“ seskupuje získané záznamy podle určitých pravidel. Používá se ve spojení s agregačními funkcemi – například s funkcí SUM(). Zmiňovaná klauzule má v rámci syntaxe příkazu SELECT své místo za klauzulí WHERE. (Lacko, 2005)

Klauzule HAVING

Podmínka umístěná za klauzulí HAVING určuje zásady výběru agregovaných záznamů. Na tomto místě musíte porozumět rozdílu mezi klauzulemi WHERE a HAVING. Klauzule WHERE slouží k omezení počtu záznamů, jež jsou později seskupovány pomocí klauzule GROUP BY. Klauzule WHERE tedy funguje nad vstupními, nikoli seskupenými záznamy. (Lacko, 2005)

Klauzule ORDER BY

Doposud jsme se nezamýšleli nad pořadím vrácených záznamů. Někdy se může dokonce stát, že data nebudou uspořádána stejným způsobem ani v případě, že dvakrát po sobě zadáme stejný dotaz. V praxi obvykle potřebujeme výpis určitým způsobem seřazený – například podle

abecedy, kalendářního data, pořadového čísla apod. záznamy můžeme třídit vzestupně (pomocí klíčového slova ASC) tj. od nejmenší hodnoty k hodnotě největší, nebo v případě třídění podle abecedy od A do Z. Můžeme třídit rovněž sestupně (DESC) od největší hodnoty po hodnotu nejmenší. K tomuto účelu se v příkazu SELECT používá klauzule ORDER BY. (Lacko, 2005)

Na následujícím obrázku je kus SQL kódu, kde je definovaná tabulka pro uživatele, který má svoje vlastnosti jako jméno a adresa, kterým se říká atributy tabulky.

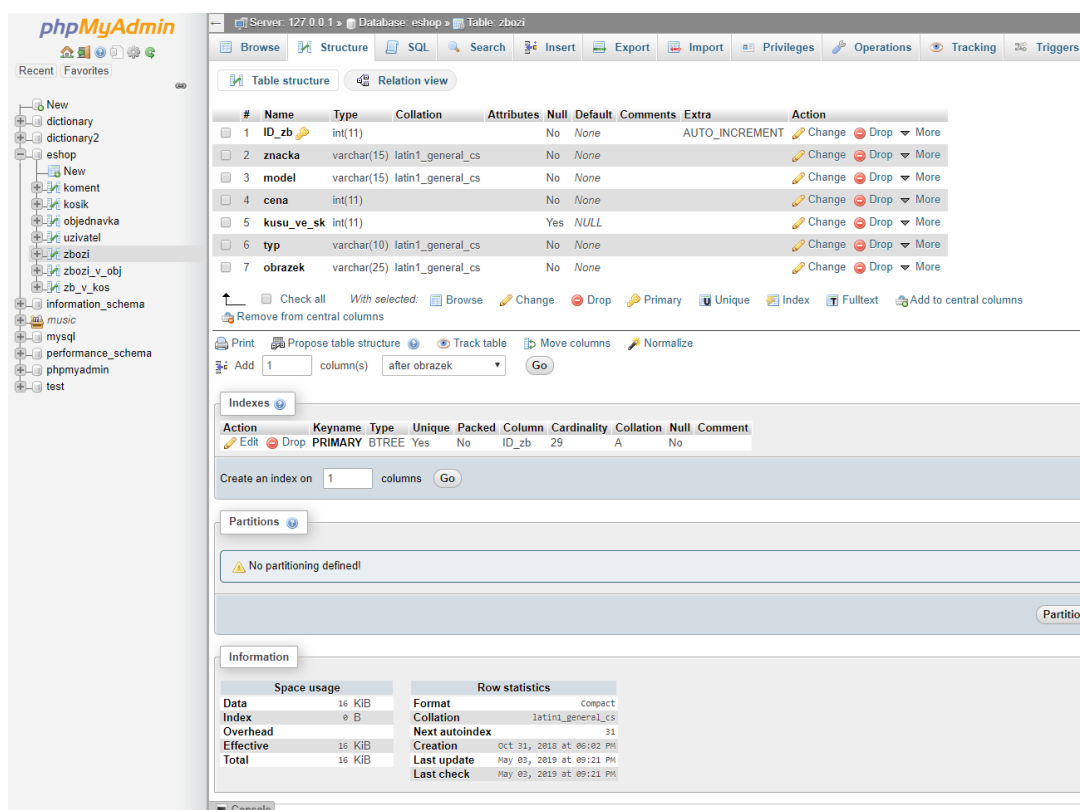
```
create table uzivatel
(
  ID_uz INT NOT NULL auto_increment primary key,
  jmeno varchar(20) NOT NULL,
  prijmeni varchar(25) NOT NULL,
  heslo varchar(20) NOT NULL,
  email varchar(30) NOT NULL,
  kraj varchar(30) NOT NULL,
  mesto varchar (30) NOT NULL,
  PSC varchar(6) NOT NULL,
  adresa varchar(40) NOT NULL
)
```

Obrázek 2.5 – SQL kód, definující novou tabulku

V prvním řádku je příkaz „create table“ po kterém následuje název tabulky. Jako první atribut je definováno identifikační číslo uživatele, datového typu INT (pro celá čísla), které se bude kvůli příkazu „auto_increment“ automaticky zvyšovat o jedničku. To znamená, že není třeba při přidávání nové položky do tabulky ručně vyplňovat toto pole. Dále se tam vyskytuje výraz „primary key“, který z daného atributu udělá primární klíč. V případě, že je atribut primárním klíčem, tak jeho hodnota musí být v každém záznamu tabulky jedinečná. Zbytek atributů je datového typu VARCHAR, určen pro text a pro každý atribut je ručně nastavená maximální délka. Příkaz NOT NULL stanovuje pravidlo, že dané pole nesmí zůstat bez hodnoty při přidávání nové položky do tabulky či aktualizování obsahu položky.

2.5.4 phpMyAdmin

Administrativní nástroj phpMyAdmin představuje webový front end k MySQL. Je napsaný v jazyce PHP a používá se pro správu obsahu databáze MySQL prostřednictvím webového rozhraní. V této bakalářské práci budou oba tyto jazyky použity k implementaci internetového obchodu. Dá se považovat za



Obrázek 2.6. Screenshot webové stránky nástroje phpMyAdmin

2.6 JavaScript

Společně s HTML a CSS tvoří JavaScript základ technologií World Wide Webu. Jde o objektově-orientovaný skriptovací jazyk, který se nejčastěji používá pro tvorbu animace a efektů v rámci webové stránky a navíc také ovládání GUI prvků jako např. tlačítka. Jeho funkce mohou být spojovány s událostmi jako pohyb/kliknutí myši, stisknutí klávesových tlačítek atd. Je na rozdíl od některých programovacích jazyků (např. PHP) interpretován webovým prohlížečem, nikoliv serverem. Autorem JavaScriptu byl programátor Brendan Eich z tehdejší firmy Netscape. V květnu roku 1995 napsal první prototyp, který byl při prvním vydání jazyku nazýván LiveScript. Jeho název byl potom změněn na JavaScript, ale nesmí se plést s jazykem Java. Shodnost názvů je výsledkem marketingového rozhodnutí firmy Netscape.

V rámci této práce se JavaScript objeví v několika místech. Nejčastěji bude použit k programování chování tlačítek a validace formulářů.

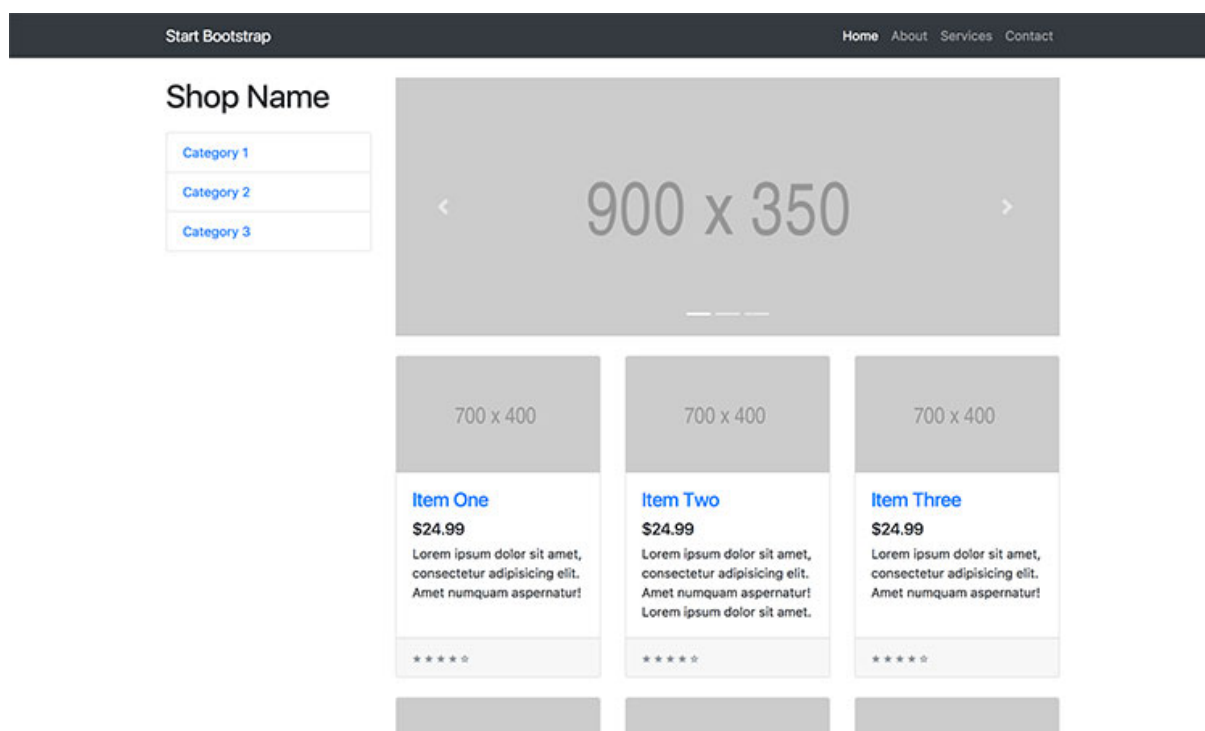
2.7 Framework a Bootstrap

V minulosti docházelo k poměrně bouřlivému zdokonalování jazyků, jak těch programovacích, tak těch značkovacích (HTML) a případně dalších, např. stylovacího jazyka CSS. Jak čas běžel, jazyky začaly být v určitém ohledu již "hotové" a nyní umožňují vytvořit

ty části aplikací, pro které jsou určené, velmi rychle a efektivně. I se sebedokonalejšími nástroji ovšem musíme stále pracovat, např. v CSS musíme pro každý náš projekt znovu a znovu definovat úplně základní styly, tedy že tabulka má mít rámeček nebo že tlačítko má být barevný obdélník. Elementy v prvních verzích jazyka HTML totiž nijak nastýlovány nebyly a kvůli zpětné kompatibilitě se to nesmí již nikdy změnit. Když již definujete ten samý rámeček pro tabulku v desátém projektu, nutně vás napadne vytvořit si nějakou knihovnu, kam si ty nepoužívanější styly uložíte. Knihovnu budete kopírovat do svých projektů a rázem zde budou styly dostupné bez nutnosti ztrácet čas jejich definováním znovu a znovu. Když je vaše knihovna dostatečně velká, tak velká, že poskytuje styly pro naprostou většinu částí současných webových stránek, říkáme ji framework. Framework je tedy ucelená knihovna nebo sada knihoven, která dává dohromady hotové řešení. A takovou CSS knihovnu vytvořili v roce 2011 ve firmě Twitter, dnes je známá jako framework Bootstrap a je nejpopulárnějším CSS frameworkem na světě. (Čapka, 2019)

Bootstrap kopíruje ty nejmodernější trendy ve webdesignu. Ty nejdůležitější jsou:

- **Responzivita** - Bootstrap styly jsou dokonale přizpůsobené pro mobilní zařízení. Je tedy 100% responzivní. A právě mobilní telefony jsou v dnešní době na webu již častější zařízení, než klasické počítače. Určitě nechcete, abyste přišli o zákazníky jen proto, že někdo otevřel vaše stránky na mobilním telefonu a písmo je tak malé, že jej nemůže přečíst.
- **Mobile-first** - Framework byl ve verzi 3 kompletně přepsán, aby podporoval mobile-first přístup. Jeho kód je tak kompaktnější a podporuje dobré praktiky.
- **Flat design** - Co dnes není flat? Váš web bude vypadat svěží a in. A když se trend změní, můžete si být jistí, že Bootstrap na to zareaguje. A pokud se vám flat přeci jen nelíbí, můžete si stáhnout jakékoli jiné ze schémat, která jsou pro Bootstrap dostupná (viz dále).
- **Grid** - Obsahuje dvanáctisloupcový grid systém. Pokud již znáte výhody grid systému, víte, že ve většině případů úplně nahradí pozicování elementů na stránce, které je potom navíc pravidelné a responzivní. (Čapka, 2019)



Obrázek 2.7. Responzivní kostra pro e-shopy (Čapka, 2019)

2.8. Elektronické obchodování

Elektronické obchodování, či zkráceně e-obchod, představuje formu nákupu a prodeje zboží a služeb elektronickou cestou. V našem případě je onou elektronickou cestou internet. Na internetu existuje celá řada různých aplikací pro elektronické obchodování, včetně:

- Online obchodů prodávajících produkty, jako např. Amazon, nebo online protipólů ke kamenným obchodům.
- Online aukce, jako je eBay.
- Online služby/webové služby, jako je BaseCamp, nebo weby založené na předplatném. (Peacock, 2011)

2.8.1 Přehled elektronického obchodování

Elektronický obchod dnes představuje velmi populární způsob podnikání. Pojďme se tedy podívat, kdo elektronické obchodování používá a k čemu.

eBay

Web eBay uvádí, že má přibližně 84 milionů aktivních uživatelů, kteří každou sekundu obchodují se zbožím v hodnotě 1900 dolarů. To znamená, že 84 milionů z nás používá eBay pro nákup a prodej zboží, ať už jako pravidelný zdroj příjmu anebo ve snaze něco málo si

přivydělat prodejem nechtěných nebo nepotřebných věcí povalujících se po domě. (Peacock, 2011)

eBay je elektronický obchod se sociálním aspektem, pracující na bázi online aukčního domu, který sám o sobě nic neprodává, nýbrž umožňuje komunitě uživatelům skrze web nejenom prodávat, ale také na něm nakupovat. To nejenom ilustruje popularitu elektronického obchodování, ale také, že se dají vydělat peníze zprostředkováním malého (i velkého) objemu online prodejů. (Peacock, 2011)

Amazon

Se ziskem více jak 19 miliard dolarů v roce 2008 se Amazon řadí mezi nejpobulárnější elektronické obchody na internetu. Výzkumy ze začátku roku 2009 ukázaly, že Amazon byl nejúspěšnějším prodejcem hudby a videa v Anglii. (Peacock, 2011)

Kamenné obchody

Velké a zaběhlé kamenné obchody, jako je Wal-Mart, Tesco a Borders, prodávají prostřednictvím internetu produkty, které zpravidla mají na prodejních. V případě obchodů Wal-Mart a Tesco si zákazníci často určí dobu, kdy se mají jejich potraviny dodat. Tyto obchody nabízejí kvůli pohodlí svých zákazníků také zboží, které na prodejně nemají a mohou ho snadno obstarat. V případě online prodejů prodejce neomezuje to, kolik toho mohou vystavit v prodejně, ale kapacita jejich distribučních skladů. (Peacock, 2011)

Menší kamenné obchody používají online prodej jako způsob, jak oslovit svými produkty širší publikum, bez omezení v podobě fyzické přítomnosti produktů. (Peacock, 2011)

Společnosti poskytující služby

Společnosti, jako je 37signals, vytváří online aplikace (jako jsou nástroje Project Management a BaseCamp) na bázi měsíčního předplatného. Mezi další příklady takovéhoto webu patří rozsáhlé weby pro distribuci souborů (dovolující „odesílat“ velké soubory e-mailem s využitím webu třetí strany) a příplatkové funkce některých webů jako např. Get Satisfaction. (Peacock, 2011)

3. Analýza požadavků a dosaženého stavu

V této kapitole se podíváme na několik hudebních e-shopů, jejich klady a zápory, a věci, které bychom mohli udělat líp. Poté budou stanoveny požadavky na náš e-shop, tak aby je mohl konkurovat.

3.1 E-shop „Bontonland“

Jeden z nejznámějších českých hudebních e-shopů, se kterým mám i já zákaznickou zkušenost. Nabízí výrobky různých typů, ale má velmi bohatou databázi studiových alb všech možných žánrů.

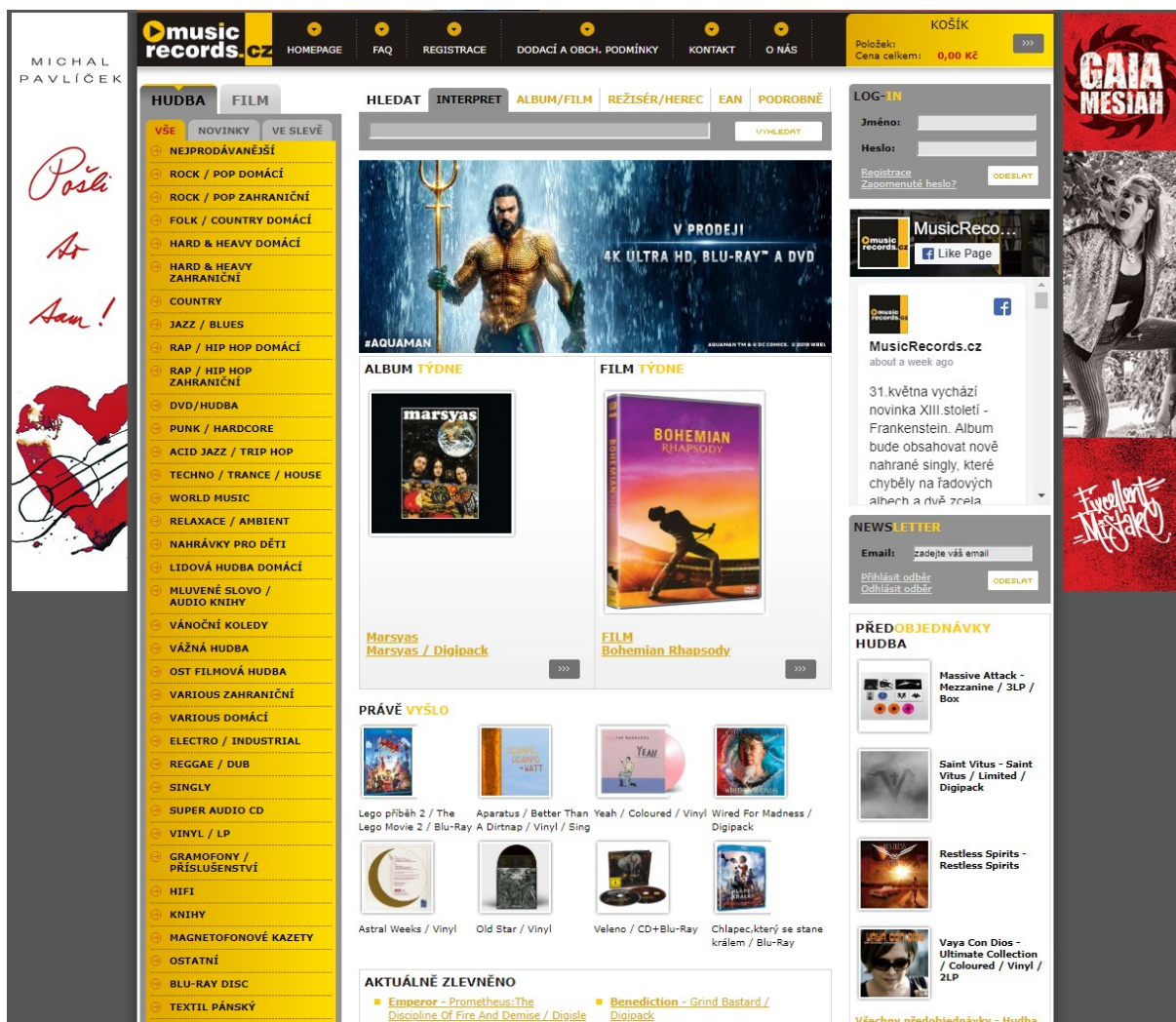


Obrázek 3.1. Vzhled stránky e-shopu Bontonland (zdroj: <https://www.bontonland.cz/hudba/>)

Orientace na tomto webu je poměrně snadná, navigace je intuitivní a neměl jsem problém rychle najít konkrétní zboží. Jelikož firma otevřela i kamenné prodejny ve větších městech, tak na svém webu nabízí náhled dostupnosti každého zboží. Z pohledu zákazníka toto velmi oceňuji, neboť často volím možnost osobního nákupu na prodejně, ale vždy předem zkontroluji dostupnost zboží na stránkách e-shopu. Na druhou stranu ale, zde chybí možnost hodnocení zboží, které pro některé klienty hraje roli při rozhodování o nákupu.

3.2 E-shop „Music Records“

Music Records, z počátku působící pouze ve formě kamenné prodejny, byl pravděpodobně jeden z prvních internetových e-shopů věnující se hudbě na českém trhu. Vzhledově nepůsobí nijak ohromujícím dojmem. Je ale dodržené standartní umístění jednotlivých prvků, tam kde je uživatel zvyklý je hledat. Na druhou stranu tento web disponuje obrovskou databází, jde o tisíce nabízených alb. V tomto ho pravděpodobně může málokdo na českém trhu předčit.

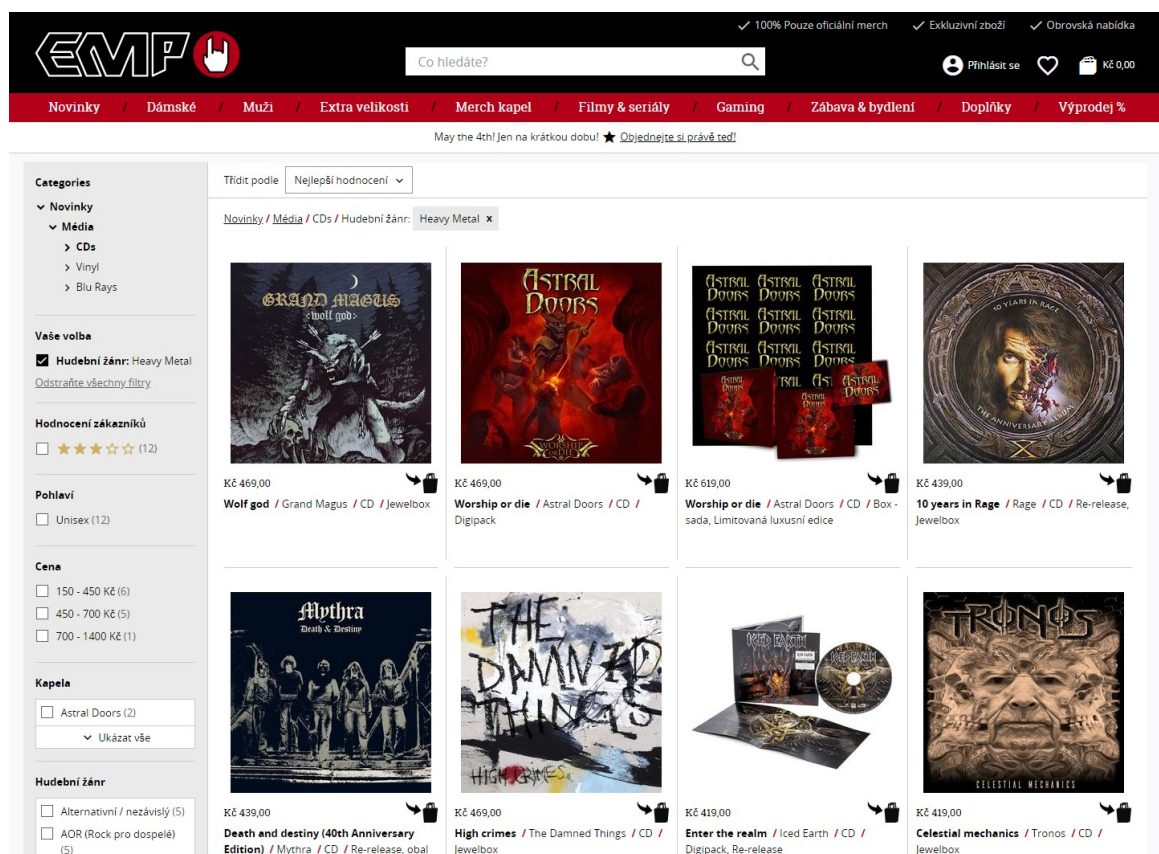


Obrázek 3.3 Náhled domovské stránky webu (zdroj: <http://www.musicrecords.cz/index.php>)

Některé možnosti jako hodnocení zboží či kontrola dostupnosti v jednotlivých prodejnách, zde chybí. Nicméně vzhledem k tomu, že je konstrukce a funkční možnosti webu jednodušší, přináší jednu velkou výhodu, a tou je rychlost načítání. V případě, že je loading stránky příliš zdlouhavý, našeho zákazníka to také může odradit.

3.3 E-shop „EMP“

Německý internetový obchod EMP nabízí širší škálu produktů, mezi kterými i velký počet CD a DVD alb se zaměřením na rockový a metalový žánr. Mají svoje stránky přeložené do patnácti jazyků, čímž získávají zájem zákazníků z celé Evropy. Oproti Bontonlandu, který trvá spíš na rozmanité nabídce žánrů, je tady kladen větší důraz na filtrování hledání produktů.



Obrázek 3.4. Hudební sekce e-shopu (zdroj: <https://www.emp-shop.cz/novinky/m%C3%A9dia/cds/>)

Z hlediska funkcí a efektů je tento web ze všech tři nejrozsáhlejší. Je zde pro většinu produktů k dispozici několik obrázků, které se dají prohlížet v rámci slideshow. Navigace není úplně intuitivní, například diskografie je schovaná v rámci „Novinky“, kde bych nehledal jeden z hlavních typů zboží, které web nabízí. Nicméně tento web jako jediný nabízí hodnocení produktů, což je pro některé zákazníky důležité. Všechny výše zmíněné možnosti ale působí negativně na načítání stránky, takže tento web je z hlediska rychlosti načítání nejpomalejší.

3.4. Požadavky

Abychom se mohli prosadit na tomto trhu, tak je třeba vybudovat e-shop, který je konkurenceschopný a odpovídá moderním standardům pro vzhled a funkčnost. Proto byli stanovené takové požadavky

- Jednoduchá a intuitivní navigace
- Responzivní design
- Slideshow a přehled novinek na hlavní stránce
- Možnost vyhledání zboží
- Možnost filtrování zboží podle žánrů, ceny a roku vydání
- Možnost rychlého nahlížení do popisu zboží najetím myši na obrázek (v rámci prohlížení katalogu, konkurence nenabízí)
- Zobrazení hodnocení ostatních uživatelů
- Administrační rozhraní pro správu katalogu, objednávky a uživatele
- Pro registrovaného uživatele možnost objednávání a hodnocení zboží
- Odeslání objednávky na email firmy a zákazníka

Těmto požadavkům jsme se snažili co nejlépe vyhovět a vybudovat aplikaci, která by odpovídala představám firmy.

4. Návrh a implementace webové aplikace

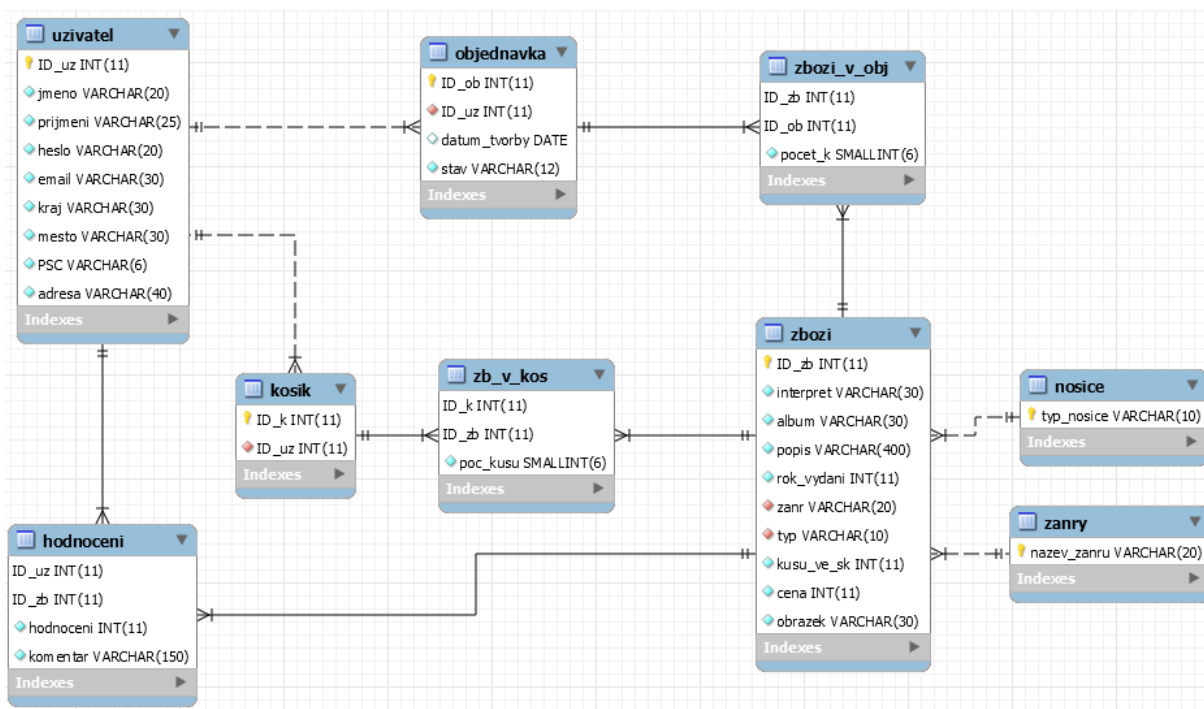
Po stanovení požadavků se můžeme začít věnovat samotné implementaci aplikace. Začneme databází a její komunikací s webovou stránkou, poté se podíváme na strukturu a funkcionalitu webové aplikace a administračního rozhraní.

4.1. Databáze

4.1.1. Relační model

V prvním kroku je třeba navrhnout relační model databáze. Pomocí tohoto modelu se dá snadněji představit fungování jednotlivých vztahů mezi entitami. Pro účely tvorby databáze (psaní kódu, manipulace s tabulkami, zobrazení relací, atd.) byl použit program MySQL Workbench. Mám s ním zkušenosti ze studia na střední škole a na univerzitě, což z něj udělalo pro mě jasnou volbu.

Na následujícím obrázku je v tomto programu zobrazen relační model databáze e-shopu.



Obrázek 4.1 - relační model databáze (zdroj: Vlastní)

4.1.2. Tabulky a vztahy

Tabulka „uzivatel“ je určena pro uživatele, který se na webu zaregistruje, aby si mohl zboží objednávat a hodnotit. Obsahuje těchto 9 atributů:

- **ID_zb** – identifikační číslo daného zboží, je primárním klíčem (tj. musí být jedinečné)
- **jmeno** – první jméno uživatele
- **prijmeni** – příjmení uživatele
- **heslo** – jeho heslo pro přihlášení do účtu
- **email** – jeho e-mailová adresa, která bude hrát role přihlašovacího jména
- **kraj** – název kraje, do kterého jeho město patří
- **mesto** – město, ve kterém uživatel bydlí
- **PSC** – poštovní směrovací číslo
- **adresa** – název ulice a číslo popisné

Tabulka „zboží“ je určena pro popis produktů, které budou na webu nabízeny.

Obsahuje těchto 10 atributů:

- **ID_uz** – identifikační číslo daného uživatele, je primárním klíčem
- **interpret** – název interpretu
- **album** – název albumu tohoto interpreta
- **popis** – popis albumu v několika větách
- **rok_vydani** – rok, ve kterém bylo dané album vydané
- **zanr** – číslo žánru, do kterého dané album patří
- **typ** – pro stanovení typu nosiče
- **kusu_ve_sk** – počet kusů daného zboží ve skladě
- **cena**
- **obrazek** – obsahuje cestu k souboru s obrázkem daného albumu

Tabulka „hodnoceni“ je určena pro vyjádření názorů uživatele na dané zboží. Její atributy jsou:

- **ID_uz** – identifikační číslo daného uživatele, je cizím klíčem
- **ID_zb** – identifikační číslo hodnoceného zboží, je cizím klíčem
- **hodnoceni** – obsahuje počet hvězd, kterými uživatel hodnotí zboží jako ve škole (známka mezi 1 a 5)
- **komentar** – místo pro slovní vyjádření názoru uživatele na dané zboží

Tabulka „kosik“ je určena pro instance nákupního košíku, který je přiřazen každému novému uživateli. Obsahuje pouze identifikační číslo košíku, což je primární klíč tabulky a identifikační číslo uživatele (cizí klíč).

Tabulka „zb_v_kos“ reprezentuje zboží, které si uživatel přidal do svého nákupního košíku. Má pouze tři atributy:

- **ID_k** – identifikační číslo nákupního košíku, je cizím klíčem
- **ID_zb** – identifikační číslo daného zboží, je cizím klíčem
- **poc_kusu** – počet objednávaných kusů daného zboží

Tabulka „objednavka“ reprezentuje objednávku, kterou uživatel vytvořil. Obsahuje pět atributů:

- **ID_ob** – identifikační číslo objednávky, je primárním klíčem
- **ID_uz** – identifikační číslo daného uživatele, je cizím klíčem
- **datum_tvorby** – datum vytvoření objednávky
- **stav** – pro zapsání stavu objednávky (zaplaceno/nezaplaceno)

Tabulka „zbozi_v_obj“ obsahuje zboží, které je obsaženo v dané objednávce. Její atributy jsou:

- **ID_ob** – identifikační číslo objednávky, je cizím klíčem
- **ID_zb** – identifikační číslo zboží které je součástí objednávky, je cizím klíčem
- **pocet_k** – počet objednávaných kusů daného zboží

Zbývají tři tabulky určené pro žánry a typy nosičů („zanry“, „nosice“). Hrají role číselníků, protože právě tyto dva atributy budou v tabulce „zbozi“ častokrát mít opakující se hodnotu. Obsahují jediný atribut, což je název a on hraje roli primárního klíče, aby byla zachována jedinečnost jednotlivých položek tabulky.

Tabulka „uzivatel“ má primární klíč ID_uz (identifikační číslo uživatele, označeno žlutým klíčem), který zajišťuje jedinečnost každého nového uživatele. Tento atribut je zároveň také cizím klíčem v tabulkách objednávka, hodnocení a košík. To bylo tak nastaveno, jelikož je nutné, aby každý výskyt nového objektu v těchto tabulkách měl kromě svého identifikačního čísla, také i svého vlastníka. V momentě, kdy se nový uživatel zaregistruje, tak vznikne nový záznam v tabulce „uzivatel“ a to taky i v tabulce „kosik“, čímž vznikne nový nákupní košík, který bude na jeho účet připojen právě přes identifikační číslo uživatele, které je v této tabulce cizím klíčem.

Postup vzniku objednávky je takový, že uživatel si nejprve vybrané zboží přidá do svého košíku. Tím se v tabulce „zb_v_kos“ (zboží v košíku) vytvoří nový záznam, který bude obsahovat identifikační číslo daného uživatele, identifikační číslo košíku, který mu patří a

počet kusů daného zboží, které by si chtěl objednat. Potom si uživatel může na stránce svého košíku dané zboží objednat a tím se v tabulce „objednavka“ vytvoří nový záznam, obsahující identifikační číslo objednávky, identifikační číslo uživatele, adresu dodání a stav objednávky. Zároveň se do tabulky „zbozi_v_obj“ zapíše konkrétní zboží, které se při objednávání nacházelo v košíku.

Na následujícím obrázku je kus SQL kódu, kde je definovaná tabulka „uživatel“

```
create table uživatel
(
  ID_uz INT NOT NULL auto_increment primary key,
  jmeno varchar(20) NOT NULL,
  prijmeni varchar(25) NOT NULL,
  heslo varchar(20) NOT NULL,
  email varchar(30) NOT NULL,
  kraj varchar(30) NOT NULL,
  mesto varchar(30) NOT NULL,
  PSC varchar(6) NOT NULL,
  adresa varchar(40) NOT NULL
) engine = innodb;
```

Obrázek 4.2 Ukázka MySQL kódu definující tabulku pro uživatele

4.2. Komunikace mezi databází a webovou stránkou

Pro realizaci e-shopu je potřeba mít prostředí pro vývoj PHP aplikací. Na internetu je široká volba softwarů ale naši volbou bude ten nejznámější z nich - XAMPP, který obsahuje pro nás nezbytný webový server „Apache“ a službu MySQL (zodpovědná pro administrativní nástroj „phpMyAdmin“). Další výhodou tohoto programu je, že zajišťuje lokální hosting pro účely testování funkcionality aplikace a tím pádem není potřeba hned řešit placený hosting. Hlavním důvodem použití jazyku PHP v rámci této práce je to, že umí zajistit komunikaci mezi webovou stránkou a databází. K tomu postačí jen pár řádků kódu. Ve všech částech webu, kde je potřeba něco vypsát nebo zapsat do databáze se používá proměnná „\$sql“ kam uložíme SQL dotaz a tyto čtyři funkce:

- `mysqli_connect()` – reprezentuje připojení k databázi
- `mysqli_set_charset()` – pro zajištění správné zpracování znaků typu UTF-8
- `query()` – spustí SQL dotaz
- `fetch_assoc()` – zachycuje řádek po řádku výsledky

```

$db = mysqli_connect('localhost', 'root', '', 'music_eshop');
mysqli_set_charset($db, 'utf8');
$sql = "SELECT * FROM uzivatel WHERE ID_uz = $ID_uz";
$result = $db->query($sql);
$row = $result->fetch_assoc();

```

Obrázek 4.3 – PHP kód, který vypisuje údaje o konkrétním uživateli pomocí SQL dotazu

4.3 Webové stránky

4.3.1 Responzivní design stránky

Firma požaduje, aby se obsah stránky mohl přizpůsobit velikosti obrazovky daného zařízení, ze které jsou navštěvovány. Jde tedy o responzivní design, který je zde zajištěn CSS knihovnou „Bootstrap“. S touto knihovnou pracujeme tak, že přes třídu „class“ vybereme příslušný objekt z knihovny použitím jeho přesného názvu. Můžeme ho také dle vlastních preferencí upravovat prostřednictvím klasických CSS příkazů.

4.3.2 Header a Footer

Nejprve je potřeba říct, že je důležité oddělit záhlaví a zápatí zvlášť v samostatném souboru a potom je do každé stránky vkládat pomocí PHP příkazu „include“. Hlavní důvod této činnosti je, že v případě rozhodnutí změnit něco v obsahu, tak se změny projeví hned ve všech souborech, kde jsou záhlaví a zápatí vloženy a nemusíme nový kód kopírovat do každého zvlášť. V rámci záhlaví tohoto webu se nachází responzivní navigace z knihovny Bootstrap. Zápatí je řešeno obdobným způsobem.

4.3.3 Přehled novinek na hlavní stránce

Tento požadavek byl splněn pomocí tohoto SQL příkazu:

```

SELECT ID_zb, interpret, album, typ, cena, obrazek FROM zboží ORDER BY ID_zb
DESC LIMIT 3;

```

V něm požadujeme výpis posledních tří produktů, přidávaných do tabulky zboží. Pomocí klauzule „ORDER BY“ určujeme, že položky budou seřazeny podle hodnoty identifikačního čísla zboží. Klíčové slovo DESC říká, že se tyto položky mají seřadit sestupně od nejvyššího a prostřednictvím LIMIT určíme, že chceme vypsát pouze první tři v seznamu výsledků.

4.3.4. Prohlížení zboží

Pro odlišení typu nosiče je použita jediná stránka (albums.php). V ní se SQL dotaz rozlišuje v podmínce WHERE na základě URL adresy, kde se nachází také upřesnění typu nosiče a je následně uloženo do PHP proměnné „\$typ“ tímto způsobem: „\$typ = \$_GET['typ'];“. SQL dotaz potom vypadá takhle:

```
„SELECT ID_zb, interpret, album, typ, cena, obrazek FROM zboží WHERE typ='$typ';“
```

Filtrování zboží rozšiřuje podmínku WHERE a je řešeno pomocí URL adresy. V rámci HTML elementu <a> určený pro hyperlinkové odkazy je nastaveno ať se proměnná „genre“ vyplňuje různě podle zvoleného žánru v rámci navigace.

```
<li class="dropdown">
  <a class="dropdown-toggle" data-toggle="dropdown" href="#">CD <span class="caret"></span></a>
  <ul class="dropdown-menu">
    <li><a href="albums.php?php echo '?typ=CD' ?>">Vše</a></li>
    <li><a href="albums_genre.php?php echo '?typ=CD&genre=rock' ?>">Rock</a></li>
    <li><a href="albums_genre.php?php echo '?typ=CD&genre=hip hop' ?>">Hip Hop</a></li>
    <li><a href="albums_genre.php?php echo '?typ=CD&genre=pop' ?>">Pop</a></li>
    <li><a href="albums_genre.php?php echo '?typ=CD&genre=metal' ?>">Metal</a></li>
    <li><a href="albums_genre.php?php echo '?typ=CD&genre=edm' ?>">EDM (Taneční)</a></li>
    <li><a href="ostatni.php?php echo '?typ=CD' ?>">Ostatní</a></li>
  </ul>
</li>
```

Obrázek 4.4 – navigace pro volbu žánrů, která do proměnné „genre“ uloží příslušný název žánru

Na další stránce (albums_genre.php) je do nové PHP proměnné „zanr“ uložen název žánru z URL adresy a ta proměnná je potom vložena do SQL příkazu rozšířením podmínky WHERE klíčovým slovem AND:

```
$zanr = $_GET['genre'];
```

```
"SELECT ID_zb, interpret, album, typ, cena, obrazek FROM zboží WHERE typ='$typ' and zanr='$zanr';"
```

Pro filtrování dle ceny či roku vydání je v případě každého typu nosiče použita jediná stránka a do URL adresy je uložen příslušný kus SQL příkazu dle zvoleného filtru. Ten je následně uložen do nové proměnné „filter“ a vložen do SQL příkazu, který vypisuje zboží:

```
$filter = $_GET['filter'];
```

```
"SELECT ID_zb, interpret, album, typ, cena, obrazek FROM zboží WHERE typ like '$typ' and zanr like '$zanr' $filter";
```



```

<?php echo " Seřadit podle: "; ?>
<a href="albums_filter.php<?php echo '?typ='.$typ.'&filter=ORDER BY rok_vydani DESC'?>">
    <input type="button" name="new" value="nejnovější" />
</a>
<a href="albums_filter.php<?php echo '?typ='.$typ.'&filter=ORDER BY rok_vydani ASC'?>">
    <input type="button" name="new" value="nejstarší" />
</a>
<a href="albums_filter.php<?php echo '?typ='.$typ.'&filter=ORDER BY cena ASC'?>">
    <input type="button" name="cheap" value="nejlevnější" />
</a>
<a href="albums_filter.php<?php echo '?typ='.$typ.'&filter=ORDER BY cena DESC'?>">
    <input type="button" name="expensive" value="nejdražší" />
</a>

```

Obrázek 4.5 – butony pro vyfiltrování zboží dle ceny a roku vydání (sestupně a vzestupně)

V případě potřeby kombinace určitého seřazení s nějakým žánrem, tak je URL adresa jen rozšířená trochu víc a obsahuje typ, žánr a kus kódu s ORDER BY klauzuli.

I když je použita ta samá stránka pro filtrování, je třeba pro orientaci návštěvníka, aby se její popis měnil podle použitého filtrování. Proto byly definovány 4 proměnné, se kterými se hodnota proměny „filter“ porovnává prostřednictvím podmínky IF. Funkce „strstr()“ porovnává textové řetězce uložené do těchto proměn a v případě, že se shodují, tak je definovaná proměnná „razeni“ kam se uloží příslušný popis stránky. Ta je pak vložena do nadpisu stránky.

```

$filter = $_GET['filter'];

$rok_nove="ORDER BY rok_vydani DESC";
$rok_stare="ORDER BY rok_vydani ASC";

$cena_levne="ORDER BY cena ASC";
$cena_drahe="ORDER BY cena DESC";

if(strstr($filter, $rok_nove) !== FALSE){
    $razeni="od nejnovějšího";
}

else if(strstr($filter, $rok_stare) !== FALSE){
    $razeni="od nejstaršího";
}

else if(strstr($filter, $cena_levne) !== FALSE){
    $razeni="od nejlevnějšího";
}

else if(strstr($filter, $cena_drahe) !== FALSE){
    $razeni="od nejdražšího";
}

```

Obrázek 4.6 – PHP kód, který porovnává textové řetězce a rozhoduje, jaký bude popis stránky

4.3.5. Rychlé nahlížení do popisu zboží

Tuto funkci naši konkurenti nenabízejí, avšak její hodnota pro zákazníka by mohla být významná, jelikož šetří čas. Zákazník si může zobrazovat podrobnosti o zboží bez nutnosti se přesouvat do jiné stránky a tímto získat informace o více zboží jednoduchým najetím myši na jejích obrázků. To je docíleno kombinací HTML a Javascript kódu. V HTML vlastnostech obrázku je stanovena vlastnost `data-toggle="popover"`, která je zodpovědná pro zobrazení a skrytí popisu. Následně je použita v následujícím kódu, kde je určeno, že najetím myši se okno s informacemi o zboží zobrazí nad obrázkem.

```
<script> $(document).ready(function() {  
    $('[data-toggle="popover"]').popover({  
        placement: 'top',  
        trigger: 'hover'  
    });  
}); </script>
```

Obrázek 4.7. Javascript kód pro rychlé zobrazení informací o zboží

4.3.6. Vyhledání zboží

V záhlaví stránky je možnost vyhledání zboží. Z textového pole se vezme hledané slovo či výraz a uloží se do URL adresy. Následně jsou v SQL dotazu vyfiltrovány všechny položky, kde se dané slovo/výraz vyskytuje v rámci názvu interpreta či názvu albumu. Pro hledání výskytu výrazu v rámci obsahu nějakého atributu se používá znak `%` na začátku a na konci hledaného výrazu.

```
"SELECT ID_zb, interpret, album, typ, cena, obrazek FROM zbozi WHERE interpret like  
'%$search%' OR album like '%$search%'";
```

Následně se stejným způsobem jak je popsáno výše v této podkapitole dají seřadit výsledky vyhledání dle ceny a roku vydání.

4.3.7. Přihlášení a registrace

Soubor „registration.php“ reprezentující stránku pro registraci nového uživatele má v sobě formulář pro vyplnění všech údajů potřebných ke zpracování objednávek a přihlašování k účtu. Pro potřeby kontroly zadávaných informací bylo potřeba pomocí javascriptu naprogramovat chování této stránky. Byla vytvořena funkce „checkForm()“, která obsahuje velké množství podmínek IF, a vyplněné údaje budou přijaty jen v případě, že jsou všechny

podmínky splněné současně. Například porovnávání zadaného hesla a opakovaná verze hesla je řešeno takto:

```
if(form.heslo.value !== form.pheslo.value) {  
    alert("Error: Hesla se neshodují!");  
    return false;  
}
```

Obrázek 4.8 javascript kód pro porovnání obsahu dvou polí (heslo a opakované heslo)

V případě splnění všech podmínek formuláře se jako poslední zkontroluje i jedinečnost zadané emailové adresy, která slouží jako identifikace uživatele. Pokud uživatel zadal jedinečnou emailovou adresu, spustí se SQL příkaz, který všechny zadané údaje vloží do databázové tabulky „uzivatel“. Hned poté druhý SQL příkaz vytvoří pro nového uživatele také vlastní košík. Pro jeho provedení se používá tzv. vnořený dotaz, který potřebujeme, aby se v rámci hodnot „values“ zobrazilo identifikační číslo nového uživatele. Jelikož nejde odhadnout jaké to číslo je, tak je třeba identifikační čísla seřadit sestupně a zobrazit pouze první výsledek.

```
$db = mysqli_connect('localhost','root','','music_eshop');  
mysqli_set_charset($db, 'utf8');  
if(isset($_POST['register']))  
{  
    $jmeno = ($_POST['jmeno']);  
    $prijmeni = ($_POST['prijmeni']);  
    $heslo = ($_POST['heslo']);  
    $email = ($_POST['email']);  
    $kraj = ($_POST['kraj']);  
    $mesto = ($_POST['mesto']);  
    $PSC = ($_POST['psc']);  
    $adresa = ($_POST['adresa']);  
  
    $sql = "INSERT into uzivatel (jmeno,prijmeni,heslo,email,kraj,mesto,PSC,adresa) "  
        . "values ('$jmeno','$prijmeni','$heslo','$email','$kraj','$mesto','$PSC','$adresa')";  
    $db->query($sql);  
  
    $sql2 = "INSERT into kosik (ID_uz) values ((SELECT ID_uz from uzivatel ORDER BY ID_uz DESC LIMIT 1));";  
    $db->query($sql2);  
    echo '<script language="javascript">';  
    echo 'alert("Registrace proběhla úspěšně!");';  
    echo '</script>';  
}
```

Obrázek 4.9. PHP kód, který zkopíruje vyplněné údaje do tabulky „uzivatel“ a poté vytvoří košík pro nového uživatele

Stránka pro přihlášení je z hlediska obsahu nejkratší, jelikož obsahuje pouze nadpis, jediný odkaz, formulář a tlačítko. Při kliknutí na tlačítko „přihlásit“, je aplikace přesměrována do souboru „login_process.php“, kde se zkontrolují přihlašovací údaje a v případě, že odpovídají existujícímu uživateli z naší databáze, tak je zahájen SESSION. Načte se verze webové aplikace pro přihlášeného uživatele („registered/index.php“) a v záhlaví aplikace se

nachází proměnná „ID_uz“. Ta si z databáze uloží identifikační číslo uživatele a je nezbytná pro všechny akce, které má uživatel možnost provádět v rámci webu, jelikož bude použita ve všech SQL příkazech této verze webu.

```
session_start();
if (isset($_POST['submit'])) {
    if (empty($_POST['email']) || empty($_POST['heslo'])) {
        echo '<script language="javascript">';
        echo 'alert("invalid email or password")';
        echo '</script>';
    } else {
        $email = $_POST['email'];
        $password = $_POST['heslo'];

        $db = mysqli_connect('localhost', 'root', '', 'music_eshop');
        $sql = "SELECT * from uzivatel where heslo='$password' AND email='$email' ";
        $result = $db->query($sql);
        if ($result === NULL) {
            echo '<script language="javascript">';
            echo 'alert("invalid email or password")';
            echo '</script>';
            header("location: login.php");
        } else if ($result == 1){
            $_SESSION['login_user'] = $email; // Initializing Session
            header("location: registered/index.php"); // Redirecting To Other Page
        }
        mysqli_close($db); // Closing Connection
    }
}
```

Obrázek 4.10 PHP kód souboru, který provádí přihlášení uživatele a zahajuje SESSION

4.3.8. Zobrazení podrobnosti o zboží

Stránka s podrobnostmi o konkrétním zboží (soubor info.php) má uloženo v URL adrese identifikační číslo zvoleného zboží a to je poté použito v rámci podmínky WHERE když se vytahují z databáze všechny informace o tomto zboží. Obrázek albumu je ve zmenšené podobě 280 na 280 pixelů a je zobrazován pomocí kombinací HTML tagu a krátkého PHP kódu, který upřesňuje cestu k souboru vypsáním obsahu pole „obrazek“ z databazové tabulky „zbozi“:

„" style="width:100%;max-width:280px">“

Při klidnutí na obrázek se zobrazí jeho přiblížená podoba, která je schovaná jako modální objekt a pomocí javascript kódu je řešeno jeho zobrazení a skrývání.

```

<script>
// Get the modal
var modal = document.getElementById("myModal");
// Get the image and insert it inside the modal - use its "alt" text as a caption
var img = document.getElementById("myImg");
var modalImg = document.getElementById("img01");
var captionText = document.getElementById("caption");
img.onclick = function(){
    modal.style.display = "block";
    modalImg.src = this.src;
    captionText.innerHTML = this.alt;
}
// Get the <span> element that closes the modal
var span = document.getElementsByClassName("close")[0];

// When the user clicks on <span> (x), close the modal
span.onclick = function() {
    modal.style.display = "none";
}
</script>

```

Obrázek 4.11. Javascript kód, který řeší akce s kliknutím na obrázek pro zvětšení

4.3.9. Přidání zboží do košíku

Na stránce „info.php“ se také nachází i možnost přidání zboží do košíku. V případě, že návštěvník není přihlášený, tak při pokusu o přidání zboží do košíku je mu pomocí javascriptu nahlášeno, že je třeba se přihlásit a automaticky se načte stránka pro registraci. V případě, že uživatel přihlášený a pokusí se zboží přidat do košíku, tak se ze sessionu vezme jeho identifikační číslo a použije se v SQL dotazu:

```
"INSERT into zb_v_kos (ID_k,ID_zb,poc_kusu) values ('$ID_k','$id','$mnozstvi');"
```

Následně je mu nahlášeno pomocí javascriptu, že zboží bylo přidáno do košíku:

```
„echo 'alert("Zboží bylo přidáno do košíku");“
```

4.3.10. Hodnocení zboží

Na stránce „info.php“ je ve spodní části zobrazen navíc i přehled hodnocení ostatních uživatelů. Za použitím identifikačního čísla daného albumu jsou z databáze hodnocení vytažena tímto SQL dotazem:

```
"SELECT      hodnoceni.hodnoceni,      uzivatel.jmeno,      uzivatel.prijmeni,
hodnoceni.komentar      FROM      hodnoceni      INNER      JOIN      uzivatel      ON
hodnoceni.ID_uz=uzivatel.ID_uz WHERE hodnoceni.ID_zb = $id;"
```

Pro zobrazení počtu hvězd byla použita bootstrapová ikona pro hvězdu:

```
"<span class="fa fa-star"></span>"
```

a PHP kód s podmínkami, které kontrolují počet přiřazených hvězd danému zboží, čímž určuje, kolik z nich bude žlutých.

Pokus o hodnocení v případě nepřihlášeného uživatele je řešen obdobně jako pokus o přidání zboží do košíku. Pokud je ale uživatel přihlášený, tak je hodnocení ve formě krátkého formuláře, kde vyplní počet hvězd (čím víc, tím pozitivněji vyhodnotil zboží) a komentář k tomu. SQL příkaz pro přidání nového komentáře do tabulky „hodnoceni“ je následující:

```
"INSERT into hodnoceni (ID_uz,ID_zb,hodnoceni,komentar) values ('$ID_uz','$id','$stars','$comment');"
```

4 3.11. Košík uživatele

Přihlášený uživatel má v záhlaví stránky tlačítko pro zobrazení svého nákupního košíku. Stránka košíku slouží také jako stránka profilu daného zákazníka. Je v ní tabulkový přehled košíku prostřednictvím HTML tabulky (tag <table>). Informace v ní je vytažena z databáze pomocí tohoto SQL dotazu:

```
"SELECT zboží.ID_zb, zboží.interpret, zboží.album, zboží.typ, zb_v_kos.poc_kusu from kosik INNER JOIN zb_v_kos on kosik.ID_k = zb_v_kos.ID_k INNER JOIN zboží on zboží.ID_zb=zb_v_kos.ID_zb where zb_v_kos.ID_k = $ID_k;"
```

Tento dotaz je delší, neboť v tabulce jsou zobrazovány sloupce ze třech různých tabulek databáze. To je docíleno prostřednictvím příkazu INNER JOIN, který spojuje tabulky na základě společného sloupce. Právě zde se projevuje role primárních a cizích klíčů. PHP proměnná „\$ID_k“ uchovává v sobě hodnotu primárního klíče košíku, který patří současně přihlášenému uživateli.

Dále se na této stránce nachází přehled fakturačních údajů uživatele. Ty se používají pro zpracování objednávky, a proto se na stránce nachází možnost si je upravit v případě, že uživatel chce, aby mu objednávka byla doručena na jinou adresu.

4 3.12. Objednávka a e-mail

Přihlášený uživatel si může prostřednictvím svého nákupního košíku zvolené zboží objednat. Když uživatel klikne na možnost „Objednat“ v tabulce košíku, aplikace je přesměrována k souboru „order.php“, který vykonává čtyři akce. Tou první je na základě email adresy přihlášeného uživatele vytáhnout z databáze informace o něm (jméno, příjmení a

adresu). Druhou akcí je do databázové tabulky „objednavka“ zapsat nově vzniklou objednávku. Další akcí je do databázové tabulky „zbozi_v_obj“ zapsat konkrétní zboží, které k dané objednávce patří společně s počtem objednávaných kusů. Poslední akce v rámci této stránky je odeslání emailové zprávy uživateli s informacemi o objednávce a pokyny k zaplacení. To je docíleno prostřednictvím PHP funkce mail, kde je kromě adresy příjemce, předmětu a obsahu zprávy důležité umístit do záhlaví tento výraz: "Content-Type: text/html; charset=UTF-8". On specifikuje znakovou sadu zprávy, která má být UTF-8, aby se správně zobrazili diakritiky českých písmen. Na konci stránky je krátký javascript kód, který zobrazí zprávu uživateli o zpracování objednávky s pokynem ke zkontrolování emailové pošty a následně je aplikace přesměrována zpátky na stránce nákupního košíku uživatele.

V rámci této bakalářské práce byla webová aplikace vytvářena na lokálním hostingu pomocí programu XAMPP, který má extention pro odesílání emailových zpráv prostřednictvím Gmailu. Je to ale potřeba ručně nastavit úpravou obsahu v programových souborech „php.ini“ a „sendmail.ini“. Stačí specifikovat Gmail jako SMTP službu, číslo portu a autentifikační údaje našeho emailu ze kterého bude XAMPP posílat zprávy o objednávkách. Pokyny k tomu se nachází v souboru „pokyny k instalaci“, který je součástí příloh této práce.

4.4 Administrační rozhraní

Aplikace obsahuje administrační rozhraní určené k úpravě obsahu databáze správcem obchodu. Slouží jak k přidávání a mazání nových položek, tak i k editaci již vložených záznamů. Obsahuje pět hlavních modulů:

- Zboží – přehled zboží a možnost přidání, úpravy a mazání
- Nosiče – přehled typů nosičů a možnost přidání nových
- Žánry – přehled hudebních žánrů a možnost přidání nových
- Objednávky - přehled objednávek a možnost úpravy
- Uživatelé - přehled uživatelů a možnost přidání a úpravy

V rámci modulů zboží, nosiče a žánry je kromě přehledu obsahu tabulky také možnost přidání nové položky, což načte stránku s připraveným prázdným formulářem. Žánr a typ nosiče jsou představeny výběrovým záznamem, který je vždy aktuální podle obsahu databázových tabulek, tzn., že při přidání nového žánru, tak se ve formuláři automaticky zobrazí k výběru také ten nový. Na následujícím obrázku je HTML kód formuláře pro přidání nového zboží ve zkrácené formě. Pro náhled do HTML kódu v úplné podobě si stačí otevřít soubor „admin_zbozi_add.php“.

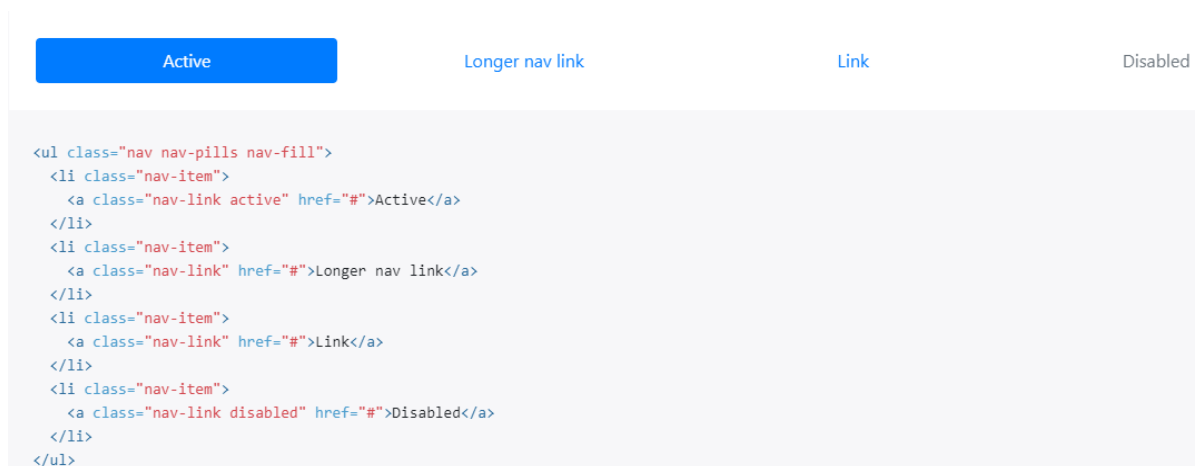
```

<form method="post">
  Interpret<br>
  <input type="text" name="interpret" placeholder="nazev interpretu" required>
  <br>Album<br>
  <input type="text" name="album" placeholder="nazev albumu" required>
  <br>Popis<br>
  <textarea rows="5" cols="50" name="popis" placeholder="popis albumu..." required></textarea>
  <br>Rok vydání<br>
  <input type="text" name="rok_vydani" placeholder="YYYY" required>
  <br>Žánr<br>
  <input type="text" name="zanr" placeholder="nazev žánru" required>
  <br>Typ<br>
  <select name="typ">
    <option value="CD">CD</option>
    <option value="DVD">DVD</option>
    <option value="LP">LP</option>
  </select>
  <br>Kusů ve skladě<br>
  <input type="number" name="kusu_ve_sk" placeholder="počet kusů" required>
  <br>Cena<br>
  <input type="number" name="cena" placeholder="- Kč" required>
  <br>Obrázek<br>
  <input type="text" name="obrazek" placeholder="img/nazev.jpg" required>
  <br>
  <input type="submit" class="button" name="pridej" value="pridej" />
</form>

```

Obrázek 4.12 HTML formulář pro přidání nového zboží do databázi

Navigace administračního rozhraní je řešena pomocí Bootstrapu, který má bohatou knihovnu navigací. Pro tuhle práci byla zvolena varianta „Fill and justify“, která využívá celou šířku webové stránky a umístí položky navigace poměrově.



Obrázek 4.13. Ukázka vzorového kódu navigace kódu navigace (zdroj getbootstrap.com/docs/4.0/components/navs/)

Způsob zobrazení obsahu databáze je ve všech modulech rozhraní obdobný:

1. Zboží – tabulkový přehled všech záznamů, které se nachází v databázové tabulce „zbozi“ pomocí SQL dotazu „SELECT * FROM zbozi“.
2. Objednávky – tabulkový přehled všech typů zboží, které se v databázové tabulce „objednavky“ nachází pomocí SQL dotazu „SELECT * FROM objednavky“.

3. Uživatelé – tabulkový přehled všech typů zboží, které se v databázové tabulce “uzivatel” nachází pomocí SQL dotazu „SELECT * FROM uzivatel“.

Admin page

zbozi								pridat zboží	
ID	Interpret	Album	Rok vydání	Zanr	Typ	Cena	Obrazek		
1	Skid Row	Slave To The Grind	1991	heavy metal	CD	250	img/slave.jpg	Edit	Delete
2	Guns N Roses	Use Your Illusion I	1991	hard rock	CD	350	img/guns.jpg	Edit	Delete
3	Eminem	The Eminem Show	2002	Hip Hop	CD	300	img/eminem.jpg	Edit	Delete
4	Kygo	Cloud Nine	2016	EDM	CD	400	img/kygo.jpg	Edit	Delete
5	Tom Odell	Long Way Down	2013	pop	CD	250	img/tom.jpg	Edit	Delete
6	Motley Crue	Dr Feelgood	1989	heavy metal	CD	350	img/feelgood.jpg	Edit	Delete
7	Dio	Holy Diver	1983	heavy metal	CD	250	img/diver.jpg	Edit	Delete
8	2Pac	All Eyez on Me	1996	hip hop	CD	250	img/2pac.jpg	Edit	Delete
9	Megadeth	Rust In Peace	1989	heavy metal	CD	250	img/peace.jpg	Edit	Delete
10	Whitesnake	Whitesnake	1987	heavy metal	CD	300	img/whitesnake.jpg	Edit	Delete
11	Metallica	And Justice For All	1988	heavy metal	CD	300	img/justice.jpg	Edit	Delete
12	Skid Row	Skid Row	1989	hard rock	CD	200	img/skid.jpg	Edit	Delete
13	Galantis	The Aviary	2017	EDM	CD	150	img/aviary.jpg	Edit	Delete

Obrázek 4.14. Náhled zboží v tabulce zboží

Ve všech modulech se v každém řádku tabulky nachází možnost úpravy (edit) a smazání (delete) dané položky. Úprava funguje přesměrováním do další stránky. V ní se nachází HTML formulář s předem vyplněnými údaji, které můžeme přepisovat. Prostřednictvím SQL příkazu UPDATE se nově zadané údaje uloží do tabulky a nahradí ty staré.

Admin page

zbozi		pridat zboží	
ID: 1			
Interpret:			
<input type="text" value="Skid Row"/>			
Album:			
<input type="text" value="Slave To The Grind"/>			
Rok vydání:			
<input type="text" value="1991"/>			
Zanr:			
<input type="text" value="heavy metal"/>			
Typ:			
<input type="text" value="CD"/>			
Cena:			
<input type="text" value="250"/>			
Obrazek:			
<input type="text" value="img/slave.jpg"/>			
<input type="button" value="uprav"/>			

Obrázek 4.15. Ukázka stránky s formulářem pro přepsání položky

Mazání je nastaveno opět na přesměrování, ale v tomto případě se jedná o prázdnou stránku, jež má za úkol položku odstranit a to na základě identifikačního čísla položky v URL adrese. Toho se docílí prostřednictvím SQL příkazu DELETE. Poté se automaticky načte předchozí stránka a vyskočí oznámení o výsledku SQL příkazu.

```
// get id value
$id = $_GET['ID'];

// delete the entry
$db = mysqli_connect('localhost','root','','music_shop');
$sql="DELETE FROM cds WHERE ID_zb= '$id' ";
$result = $db->query($sql);
```

Obrázek 4.16. Ukázka části kódu, která řeší odstranění položky

5. Závěr

Cílem mé bakalářské práce bylo vytvořit internetový obchod s administračním rozhraním pro vlastní budoucí podnik, který bude realizovat prodej diskografií interpretů a kapel různých žánrů. Jedním z hlavních požadavků bylo, aby uživatel si mohl zboží objednávat a hodnotit. Dále, aby aplikace uměla odesílat emailovou zprávu o objednávce jak klientovi, tak i firmě.

K tvorbě aplikace byly využity znalosti z některých jazyků pro navrhování webových stránek jako HTML a CSS, ale bylo potřeba taky získat znalosti z jazyku PHP, aby mohla být realizována komunikace mezi e-shopem a databází. Dále pro naprogramování chování HTML elementů stránky byl využit jazyk Javascript.

V teoretické části práce jsme se ve hloubce seznámili se všemi technologiemi, které byly v rámci implementace aplikace použity. Převážně je tam kladen důraz na CSS, který zajistí námi požadovaný vzhled webových stránek a jazyk SQL, pomocí kterého zapisujeme a vytahujeme data z databáze.

V kapitole Analýza požadavků a dosaženého stavu jsou zanalyzovány e-shopy konkurenčních firem. Je kladen důraz na prvky funkcionality těchto webů, které jsou nezbytné pro navržení moderního internetového obchodu a na základě nich byly stanoveny realistické požadavky pro náš e-shop, tak aby mohl být konkurenceschopný.

V kapitole o návrhu a implementaci aplikace je podrobně vysvětlena každá část funkcionality aplikace. Nejprve jsou objasněny obsah a tvorba databáze, která je základním kamenem pro internetový obchod. Dále je vysvětlen způsob dosažení funkcionality každé jednotlivé stránky s příloženými kusy kódu a obrázky. Na konci je znázorněné administrační rozhraní a metody jeho realizace.

Všechny požadavky webové aplikace byly splněny. Registrovaný uživatel se může přihlásit a hodnotit či objednávat si zboží. Nicméně jelikož byla realizovaná v rámci lokálního hostingu, některé funkcionality jsou omezené. Pro skutečný provoz webové stránky bude potřeba změnit způsob řešení několika služeb jako například odesílání emailových zpráv.

Seznam použité literatury

Tištěné knihy

SKLAR, David. *PHP 7: praktický průvodce nejrozšířenějším skriptovacím jazykem pro web*. Přeložil Jan POKORNÝ. Brno: Zoner Press, 2018. ISBN 978-80-7413-363-3.

WELLING, Luke a Laura THOMSON. *Mistrovství PHP a MySQL*. Přeložil Ondřej BAŠE. Brno: Computer Press, 2017. ISBN 978-80-251-4892-1.

PEACOCK, Michael. *Programujeme vlastní e-shop v PHP 5*. Brno: Computer Press, 2011. ISBN 978-80-251-3181-7.

PONKRÁC, Miloslav. *PHP a MySQL: bez předchozích znalostí*. Brno: Computer Press, 2007. ISBN 978-80-251-1758-3.

CASTRO, Elizabeth a Bruce HYSLOP. *HTML5 a CSS3: názorný průvodce tvorbou WWW stránek*. Brno: Computer Press, 2012. ISBN 978-80-251-3733-8.

Internetové zdroje

ČAPKA, David. *Úvod do CSS frameworku Bootstrap*. [online]. 2019-02-09 [cit. 2019-03-17]. Dostupné z: <https://www.itnetwork.cz/html-css/bootstrap/kurz/uvod-do-css-frameworku-bootstrap>

Seznam zkratek

CERN - Conseil Européen pour la recherche nucléaire

CSS – Cascading Style Sheets

E-R – Entity Relationship

HTML – HyperText Markup Language

MySQL – My Structured Query Language

PHP – Hypertext Preprocessor

RGB – Red Green Blue

SQL – Structured Query Language

URL – Uniform Resource Locator

WWW – World Wide Web

SMTP - Simple Mail Transfer Protocol

Prohlášení o využití výsledků bakalářské práce

Prohlašuji, že

- jsem byl seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. – autorský zákon, zejména § 35 – užití díla v rámci občanských a náboženských obřadů, v rámci školních představení a užití díla školního a § 60 – školní dílo;

- beru na vědomí, že Vysoká škola báňská – Technická univerzita Ostrava (dále jen VŠB-TUO) má právo nevýdělečně, ke své vnitřní potřebě, bakalářskou práci užít (§ 35 odst. 3);

- souhlasím s tím, že bakalářská práce bude v elektronické podobě archivována v Ústřední knihovně VŠB-TUO a jeden výtisk bude uložen u vedoucího bakalářské práce. Souhlasím s tím, že bibliografické údaje o bakalářské práci budou zveřejněny v informačním systému VŠB-TUO;

- bylo sjednáno, že s VŠB-TUO, v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu § 12 odst. 4 autorského zákona;

- bylo sjednáno, že užít své dílo, bakalářskou práci, nebo poskytnout licenci k jejímu využití mohu jen se souhlasem VŠB-TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše).

V Ostravě dne 20. 6. 2019



.....
Aleksandar Nikolaev Kolev

Seznam příloh

Příloha 1: CD s webovou aplikací.

Přílohy

Příloha 1: CD s webovou aplikací.

Na přiloženém CD se nachází:

- Zdrojový kód webové aplikace
- Zdrojový kód databáze
- Pokyny pro instalaci aplikace